[CMSC 425/525] Assignment #1: Input Domain Modeling

Dr. Kosta Damevski
Due: End of Day, February 12th, 2023

The `java.util.BitSet` is an class that is part of Java's standard library that enables its users to work with bitvalues. The class has numerous methods, but, for this assignment, please only consider only the following four:

```
public BitSet(int nbits)
//Creates a bit set whose initial size is large enough to explicitly
//represent bits with indices in the range 0 through nbits-1. All bits are initially false.
//Parameters: nbits - the initial size of the bit set
//Throws: NegativeArraySizeException - if the specified initial size is negative

public void set(int bitIndex)
//Sets the bit at the specified index to true.
//Parameters: bitIndex - a bit index
//Throws: IndexOutOfBoundsException - if the specified index is negative

public boolean get(int bitIndex)
//Returns the value of the bit with the specified index. The value
//is true if the bit with the index bitIndex is
//currently set in this BitSet; otherwise, the result is false.
//Parameters: bitIndex - the bit index
//Returns: the value of the bit with the specified index
//Throws: IndexOutOfBoundsException - if the specified index is negative

public void flip(int bitIndex)
//Sets the bit at the specified index to the complement of its current value.
//Parameters: bitIndex - the index of the bit to flip
//Throws: IndexOutOfBoundsException - if the specified index is negative
```

Complete the following tasks for testing the capabilities of the four methods of the `BitSet` class using input space partitioning. Assume that Java11 is being used.

1. Devise a set of characteristics based on the functionality described in the four methods of the `BitSet` class. Aim for at least one interface-based and one functionality-based characterstic. Document the characteristics and their corresponding blocks in a table. Make sure that each of the blocks is disjoint and that they together cover the entire input domain.
2. Devise a set of test requirements based on the characteristics (blocks) from above, using Base Choice Coverage, documenting the base case and any unfeasible combinations. Again clearly specify all of the requirements in a table.
3. Implement a set of tests that cover all of the feasible test requirements. Add a comment to the top of each test that indicates the test requirement(s) that are covered.

Turn in your code and all associated tables and description in Canvas.